



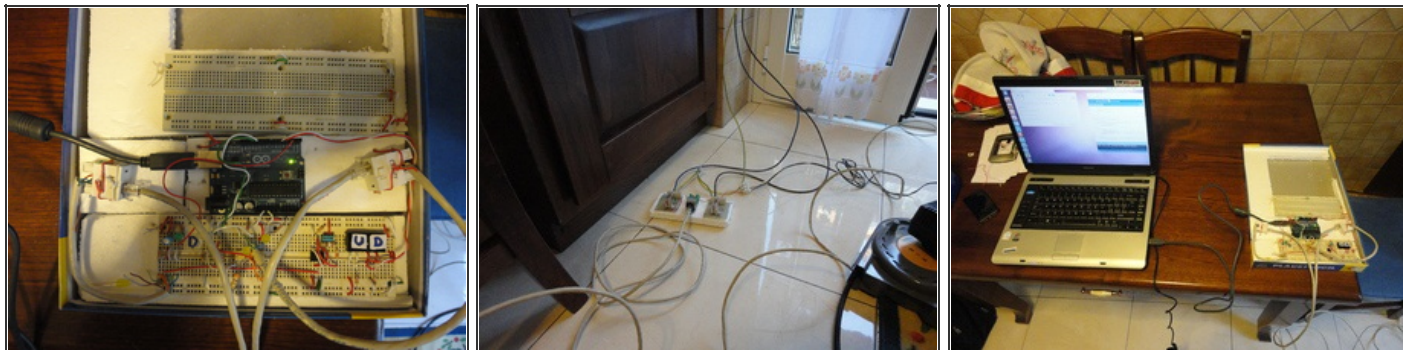
# Automatic Shutters

Written By: robertodelle

## SUMMARY

The purpose of this project is to control an electric roller shutter using an Arduino Uno. The prototype that I made is my first electronic project with an Arduino and I hope this instructable and the solutions that I found are useful for your own electronics projects. My aim was to transform an electric roller shutter into an automatic roller shutter. A possible extension of this project could be a complex automated system for a house. A central microcontroller (another Arduino, perhaps) could control all of the house's electric roller shutters; this would enable coordinated operation at different times of day. This project is limited to a prototype; after so many years I wanted to return to building electronic circuits.

## Step 1 — Automatic Shutters



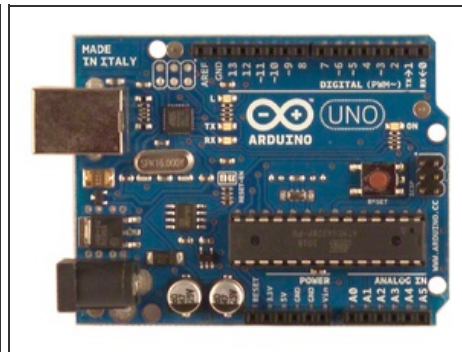
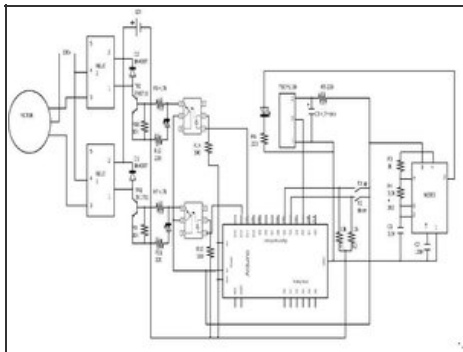
- Functions: 1. Open and close the shutter when the appropriate button is pressed; 2. Stop the shutter when it is completely open or closed; 3. Stop the shutter when it is moving down and the "up" button is pressed; 4. Stop the shutter when it is moving up and the "down" button is pressed.
- Here are some videos of the device in operation:
  - <http://www.youtube.com/embed/DPf4oQDOABU>
  - <http://www.youtube.com/embed/ywhf1mCyTkg>

## Step 2



- Principle of operation: The Arduino counts the holes on the roller shutter using an infrared receiver; the infrared emitter diode is positioned on the other side of the roller shutter. When a hole is sensed, the Arduino increments or decrements a counter to determine the position of the roller shutter. To connect the actuators and the sensor to the Arduino I used two LAN cables (see last step).

### Step 3



- To draw the schematic I used the Dia program. It is very simple to learn, available for Linux and Windows, and it is freeware. You can download it [here](#). The NE555 is configured as an astable multivibrator which generates a square wave at 38kHz to drive the infrared emitter. The infrared receiver is a TSOP1138 with an internal filter for 38 kHz, and the demodulated output signal can be directly connected to a microprocessor. For the safety of the Arduino I connected its outputs to two optoisolators (but this is not required).
- Finally, I made two actuators using two old 14v relays and a few other components such as two 2N1711 transistors, two diodes and a 12v power supply.
- Here is the schematic: <https://docs.google.com/open?id=0B8tHAfj...>

### Step 4

```

sketch_jan10a [Arduino 0022ubuntu0.1]
File Edit Sketch Tools Help

sketch_jan10a.ino
// constantes
const int sm = 2; // movement sensor
const int buttonPin_13 = 0; // button up
const int buttonPin_12 = 4; // button down
const int ledPin = 13; // LED for check buttons
const int d1 = 11; // motor UP
const int d0 = 12; // motor DOWN

// variables
int ContatoreSensoreMovimento = 0; // counter move sensor
int Contatore_t3_t2 = 0;
int RegistroOutputd0 = 0;
int RegistroOutputd0 = 0;

// variables sm
int State_sm = 0; // state
int lastState_sm = 0; // last state
// variables t3
int ButtonState_t3 = 0; // state
int lastButtonState_t3 = 0; // last state
// variables t2
int ButtonState_t2 = 0; // state
int lastButtonState_t2 = 0; // last state
}

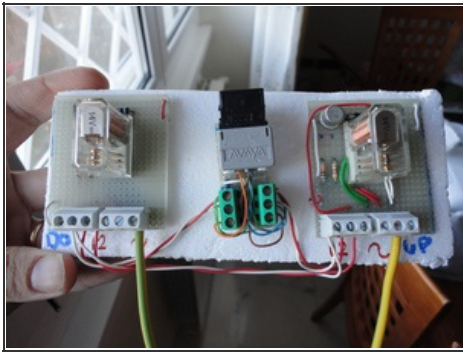
Done compiling.
Binary sketch size: 3176 bytes (of a 32256 byte maximum)
159

```



- Here is the Arduino code: <https://docs.google.com/document/d/1-nug...>

## Step 5



- Other photos of this project.

This document was last generated on 2012-11-03 03:28:41 AM.